

### REMARKS/ARGUMENTS

Claims 1-28 are pending in this application. Claims 1-28 stand rejected. Claims 1, 10, and 19 have been amended to clarify the claimed subject matter. No new matter has been added. In view of the following remarks, reconsideration and allowance of all pending claims are respectfully requested.

#### **Information Disclosure Statement**

Applicants note the dates for two references could not be determined by the Examiner. Applicants are submitting herewith a revised IDS to clarify the undetermined dates.

#### **The Specification**

Applicants have updated the status of pending parent priority applications.

The specification of the instant application was objected to because a computer program listing contains more than 300 lines. Applicants have redacted the code and stored the code on a CD ROM. The CD file has been incorporated by reference by a paragraph added near the beginning of the amended specification. Applicants believe the objection to the amended specification should be removed.

### **Claims Objections**

Claims 1, 4, 5, 10, 13, 14, 19, 23, and 24 are objected to because the term “ML” is not recognized by the Examiner as a standard abbreviation for that term. Applicants have amended the independent claims to define “ML” as “markup language.”

Claim 19 was objected to because of a repeated phrase. The claim has been amended to remove the repeated phrase.

### **Claim Rejections under 35 U.S.C. § 101**

The Office Action rejected claims 1-9, and 19-26 under 35 USC § 101 because the claimed invention is directed to non-statutory subject matter. The Office Action also states that there is no functional relationship imparted by the data (the claimed computer-executable components) to a computing device and that the claim describes software per se recorded on memory.

Applicants respectfully traverse the rejections. Beauregard-type claims are clearly patentable subject matter. Examination guidelines for patentable subject matter describe “functional descriptive material” consisting of data structures and computer programs which impart functionality when employed as a computer component. In determining whether the claim is for a “practical application,” the focus is not on whether the steps taken to achieve a particular result are useful, tangible and concrete, but rather that the final result is “useful, tangible and concrete.”

Generally, it is appreciated that users often record software per se on computer-readable media because the software provides results that are useful, tangible and concrete. More specifically, one such possible result is efficiently storing native document settings in XML that are human-readable, which simplifies maintenance of the document by humans, and thus provides useful, tangible, and concrete results. Notwithstanding the above, applicants have amended the claims to more clearly specify claimed subject matter.

#### **Claim Rejections under 35 U.S.C. § 102(b)**

The Office Action rejected claims 1-26 under 35 USC § 102(b) as being anticipated by Altova Inc. & Altova GmbH, "XML Spy 4.0 Manual," pages 1-401 ("XML Spy"). The Office Action argues claims 1-9 recite similar subject matter to that specified in claims 10-18, and that claims 10-18 are more clearly expressed and are used as exemplary claims.

With regard to claim 10, XML spy fails to teach or suggest editing an electronic document comprising automatically generated document properties, prompting and receiving custom properties for the electronic document from a user, and encoding in a markup language (ML) format the electronic document, the automatically generated document properties, and the custom properties received from the user.

To aid the Examiner in understanding of the claim, attention is drawn to an example of document properties that is given starting on page 7 of the specification. For example, many applications used to edit objects within a document are capable of maintaining certain pieces of information about a document. Some of the pieces are automatically generated by the

application and some are created by the users. Those pieces of information are not usually considered part of the document (they are not typically visible in the document body and there is special user interface to control them), but rather are options that help the application manage the document. The document property settings are represented in an ML file (such as XML).

An example list of automatically saved document property settings can include LastAuthor, Revision, TotalTime, Created, LastSaved, and the like. These properties are clearly properties that are, for example, monitored, tracked, and saved by the application, and are not normally maintained by the user, nor are they normally presented when a user tries to edit the document per se.

In contrast, the cited art shows editing capabilities for elements in the document itself. For example, page 5 and 6 of XML Spy shows an enhanced grid view of an XML document that allows a user to see and directly manipulates element in the XML document. The example shows actual data the document contains (an invoice list with prices for individual elements). Thus, the cited art teaches editing XML elements in the document itself (that are normally displayed as part of the document) and does not teach or suggest maintaining properties about a document, either manually or automatically. Pages 11 – 17 similarly show editing of elements with in a document per se.

Accordingly, the cited art fails to teach or suggest automatically or manually editing document properties. Claim 10 is allowable.

Regarding claim 11, the cited art fails to teach or suggest, for example, editing an electronic document comprising automatically generated document properties as discussed above. Claim 11 is allowable.

Regarding claim 12, the cited art fails to teach or suggest, for example, editing an electronic document comprising automatically generated document properties as discussed above. Claim 12 is allowable.

Regarding claim 13, XML spy fails to teach or suggest backwards compatibility properties in response to a version number of the computer-executable component, and wherein the third component is further arranged to encode the backwards compatibility properties in an ML format. In contrast, the cited portion of XML Spy (pages 51-90) generally discusses updating of elements in XML documents. The example on page 70 teaches modifying an underlying schema document to update a telephone phone extension from two to three digits. This teaches away from the versioning information because the schema for document must be changed to handle data of incompatible versions, rather than (for example) changing the schema to handle versioning information in a document.

The note on page 73 of XML Spy also does not teach versioning. Instead, it appears to direct an XML Spy user to use a later version of XML Spy (containing a bug fix) when encountering a certain error.

The example starting on page 77 merely teaches importing and exporting database data. The note on page 79 teaches exporting to different versions of an Access ® database, and does

not include versioning information in the document itself. For example, to export data to an Access 97 database, it is necessary to create an empty Access® 97 database first, and then selecting the export option to choose an existing Access® database. Thus the document itself is not modified to permit backwards compatibility. Rather, the user must start over with a previous version. Claim 13 is allowable.

Regarding claim 14, the cited art fails to teach or suggest a third component that is arranged to encode the application environment properties. Instead the prior art teaches updating of elements in XML documents as discussed above. Claim 14 is allowable.

Regarding claim 15, the cited art fails to teach or suggest the automatically generated document properties that comprise a LastAuthor element. Instead the prior art teaches on page 132 logging onto password protected servers. The User and Password information (which is normally kept in a secure manner) is not kept in the XML data. Claim 15 is allowable.

Regarding claim 16, the Office Action asserts that the term “married” occurs in the disclosure solely as an example of what might be save in a “custom property setting” and that the designation “married” is read as non-functional descriptive material and not limiting of the claimed invention. Applicants traverse the assertion that the specification limits the scope of the claims based on the examples given in the specification. The term “married” has inherent functionality in that it designates the marital status of a person. Such functionality is useful in that it, for example, allows the software to determine whether certain costs, services, benefits, and the like should be accorded to a certain employee.

The cited art (pages 287-362 of XML Spy) generally describe scripting (providing a scripting environment for creating, managing, and storing forms, macros and event handlers), an API (for the built-in scripting environment of XML Spy), and a document editor (for editing an XML document based on templates created in an XSLT designer). Page 311 shows an object model that comprises a “root object” Application that maintains a list of ActiveDocuments. The list of active documents is not maintained by a document itself so ActiveDocuments is not a (Boolean) custom property that is part of the electronic document that is encoded in a markup language format.

Regarding claim 17, the cited art fails to teach backwards compatibility properties comprising a justification element. Instead the art teaches project control files (on page 147), a “convert” menu (page 198), and importing Word® documents (page 204).

The project control files are stored separately from documents within a particular project. Additionally, the note on page 147 states that the project control files should only be edited by advanced users. The convert menu merely teaches converting back and forth between text, word processor, database, and XML files. XML Spy merely teaches importing conversion of Word® ('97 or 2000-version) documents into XML if the paragraph styles have been used. The backwards compatibility elements (if any) are not stored in XML because the document imported is not in XML. Thus the cited art does not teach backwards compatibility properties comprising a justification element. Claim 17 is allowable.

Regarding claim 18, the cited art fails to teach the zoom element. In contrast, the cited art teaches on page 136 a miniature display of the document to be printed. There is no teaching

that a zoom element be comprised by the application environment properties. On page 191, the cited art teaches a zoom command to control the zoom factor of the content model view. Again, there is no teaching that a zoom element be comprised by the application environment properties. Claim 18 is allowable.

With regard to claim 19, XML spy fails to teach or suggest an electronic document file that comprises automatically generated document properties, an editor that is arranged to prompt and receive custom properties for the electronic document from a user, and an encoder that is arranged to encode in a markup language (ML) format the electronic document, the automatically generated document properties, and the custom properties received from the user.

As discussed above, the cited art fails to teach or suggest automatically generated properties and/or an editor for prompting and receiving custom document properties for the electronic document from a user. Claim 19 is allowable.

Claims 21-26 depend from claim 19, and are thereby allowable.

With regard to claim 1, XML Spy fails to teach or suggest a first component that is arranged to edit an electronic document comprising automatically generated document properties, a second component that is arranged to prompt and receive custom properties for the electronic document from a user; and a third component that is arranged to encode in a markup language (ML) format the electronic document, the automatically generated document properties, and the custom properties received from the user.



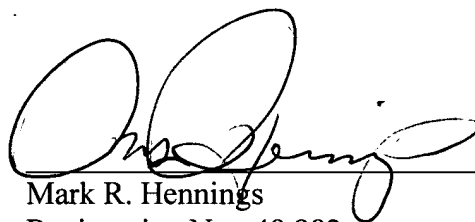
As discussed above, the cited art fails to teach or suggest automatically or manually editing document properties. Claim 1 is allowable.

Claims 2-9 depend from claim 1, and are thereby allowable.

In view of the foregoing amendments and remarks, all pending claims are believed to be allowable and the application is in condition for allowance. Therefore, a Notice of Allowance is respectfully requested. Should the Examiner have any further issues regarding this application, the Examiner is requested to contact the undersigned attorney for the applicant at the telephone number provided below.

Respectfully submitted,

MERCHANT & GOULD P.C.



Mark R. Hennings  
Registration No. 48,982  
Direct Dial: 206.342.6289

MERCHANT & GOULD P.C.  
P. O. Box 2903  
Minneapolis, Minnesota 55402-0903  
206.342.6200

